EJFA Emirates Journal of Food and Agriculture

Emirates Journal of Food and Agriculture 37: 1–14
doi: 10.3897/ejfa.2025.172982

RESEARCH PAPER SPECIAL COLLECTION NAME

Comparative evaluation of deep learning architectures for brinjal fruit disease classification

Basab Nath[1], Yonis Gulzar[2], Sagar Tamang[3], Mohannad Alkanan[2]

1 *School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India*
2 *Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa, 31982, Saudi Arabia*
3 *Centre for Educational Technology, Indian Institute of Technology Patna, Bihta, 835217, India*

Corresponding authors: Basab Nath (basab.nath@bennett.edu.in); Yonis Gulzar (ygulzar@kfu.edu.sa)

Academic editor: Yongming Liu ♦ **Received** 24 September 2025 ♦ **Accepted** 6 November 2025 ♦ **Published** 9 December 2025

Abstract

Micronutrient malnutrition, especially anaemia, remains a pressing global concern, underscoring the value of affordable and nutrient-dense crops such as brinjal (*Solanum melongena* L.). Yet, its cultivation is hampered by fruit diseases that significantly diminish both yield and market quality. Conventional diagnosis depends on manual inspection, which is time-consuming and prone to error. To overcome this limitation, we conducted a comprehensive benchmarking of contemporary deep learning architectures for automated brinjal fruit disease recognition under real-world field conditions. For this purpose, we developed the BrinjalFruitX dataset containing 3,077 images of five classes—Healthy, Phomopsis Blight, Wet Rot, Shoot and Fruit Borer, and Fruit Cracking—captured under natural variability. We evaluated ten representative convolutional neural networks (CNNs), including models from the VGG, ResNet, Inception, EfficientNet, and MobileNet families, across four training paradigms: training from scratch, transfer learning, fine-tuning, and full training ("Full Monty"). Performance was systematically analyzed using accuracy, macro- and weighted-F1 scores, training loss, confusion matrices, and computational efficiency indicators such as parameter count, FLOPs, and inference latency. Among the tested models, MobileNetV2 with Full Monty training achieved the best balance of performance and efficiency, reaching 97.98% accuracy, a macro-F1 of 0.9793, and operating with only 3.4M parameters, 0.30B FLOPs, and an inference time of 3.2 ms per image. While InceptionV3 and VGG16 also produced competitive results, they required considerably higher computational resources. In contrast, deeper ResNets and EfficientNetB0 offered inferior accuracy despite higher complexity. These findings highlight MobileNetV2 with full training as a practical and lightweight solution for on-device and farmer-oriented applications. This study establishes a strong benchmark for advancing deep learning–driven disease detection in sustainable agricultural systems.

Keywords

Brinjal fruit disease classification, convolutional neural networks (CNNs), transfer learning, model efficiency, MobileNetV2

Introduction

Micronutrient malnutrition, also known as hidden hunger is a worldwide health problem. Thirty point seven percent of women of reproductive age have anaemia, while the prevalence is even held at 35.5% among pregnant women based on most recent data released by World Health Organisation (WHO, 2025). Prevalence of anaemia among children aged 6–59 months was 39.8% in 2019. Anaemia causes fatigue, decreased productivity, and bad maternal-child health outcomes — passed on across generations as poor motor and cognitive

Copyright *Basab Nath, et al.* This is an open access article distributed under the terms of the Creative Commons Attribution License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

development (World Health Organization 2025). The world now is not on course to reach the target of cutting anemia by half by 2030.

Vegetables are a critical component of human diets, as they possess necessary vitamins, minerals, phytonutrients and dietary fiber. Yet, in most areas, vegetable intake is still less than the recommended amount (Afshin et al. 2019). Crops, like brinjal (eggplant, Solanum melongena L.) and its wild relatives are important in providing dietary diversity for combating hidden hunger. Eggplant fruits are a cheap source of calcium, phosphorus, iron and B-group vitamins. They also include polyunsaturated fatty acids that reduce cholesterol, thereby adding nutritional value (Food and Agriculture Organization of the United Nations 2022).

Brinjal is one of the globally important solanaceous vegetable crops. It covers 18.94 lakh hectares during 2022–23 with the production of 593 lakh tonnes an average yield of 31,383 kg/ha. This makes the brinjal as the fifth most valuable Solanaceae crop after potato, tomato, pepper and tobacco (Food and Agriculture Organization of the United Nations 2022). China is the dominant producer with 38.31 million tonnes (64.5% of worldwide production), followed by India with 12.76 million tonnes (21.5%). Egypt, Turkey and Indonesia are also major producers. Brinjal is one of the first five economically and socially most important vegetable crops in Asia and the Mediterranean.

In India, the annual crop production is approximately 12.7 million tonnes signifying its significant role in the rural livelihood and national supply of vegetables (Professor Jayashankar Telangana State Agricultural University 2025). Growing is mainly in places like West Bengal, Odisha, Gujarath and Madhya Pradesh. Brinjal: The total area under brinjal in 2023–24 according to the third advance estimates was 6.78 lakh hectares with a production of 128.79 lakh tonnes, a little less than last year (Food And Agriculture Organization Of The United Nations 2022). For instance in Telangana, the price forecast for Yasangi (Rabi) 2024–25 pre-harvest estimated the range of Rs. 1720–2070 per quintal, highlighting both the economic importance and vulnerability of crops to relative production and market conditions (Professor Jayashankar Telangana State Agricultural University 2025).

As for Africa, African eggplant (Solanum aethiopicum) is one of the traditional vegetables in East and West Africa. It is grown for fruits as well as leaves and has high phenotypic variation in size, colour and taste. The crop does best under hot and dry conditions and features in the local people's diet as well as farming system (Mvungi et al. 2025). Nevertheless, it has been overlooked in socioeconomic studies and commercial crop breeding schemes except being used as a resistance donor to S. melongena. Breeding efforts by the World Vegetable Center and national institutions have resulted in the development of new improved cultivars with higher yields and decreased bitter taste levels; adopting these is geographically limited (Mvungi et al. 2025).

Despite its economic and nutritional significance, brinjal cultivation faces serious challenges from fruit diseases. Common problems include shoot and fruit borer, wet rot, fruit cracking, and phomopsis blight. These diseases reduce yield, affect fruit quality, and lead to economic losses. Traditional monitoring depends on manual inspection, which is time-consuming and often unreliable. In recent years, artificial intelligence (AI) and deep learning (DL) have been introduced as effective solutions for plant disease detection (Gulzar 2025a, 2025b). However, most prior studies focus on controlled datasets of leaves, such as PlantVillage, while research on fruit disease detection remains limited. Targeted studies on brinjal fruits are scarce, and systematic evaluations of advanced architectures such as convolutional neural networks (CNNs) and vision transformers (ViTs) are yet to be reported (Professor Jayashankar Telangana State Agricultural University 2025).

This study seeks to address this gap through a comprehensive benchmark of state-of-the-art deep learning models for brinjal fruit disease classification. Special emphasis is placed on class imbalance and efficiency to support robust, real-world applications in agriculture. The major contributions of this work are as follows:

1. Development of the BrinjalFruitX dataset, a curated collection of fruit images captured under realistic field conditions with natural variability.
2. Systematic benchmarking of ten widely used CNN architectures (VGG, ResNet, Inception, EfficientNet, and MobileNet) under four distinct training strategies.
3. Comprehensive evaluation using balanced classification metrics (accuracy, macro-/weighted-F1, confusion matrices) alongside efficiency indicators (parameters, FLOPs, inference time).
4. Recommendations for deploying efficient yet accurate models for farmer-level and mobile applications.

Related work

Plant disease detection with deep learning

Deep learning has transformed plant pathology research by enabling automatic disease detection from images. Early works demonstrated the capability of convolutional neural networks (CNNs) to identify leaf diseases in crops such as sunflower (Gulzar et al. 2023), soybean (Gulzar 2024), corn (Alkanan and Gulzar 2024), and rice (Seelwal et al. 2024) with accuracies exceeding 90%. Large controlled datasets, such as PlantVillage, have accelerated progress, but their laboratory conditions limit robustness in real fields. For brinjal, where fruit symptoms occur under variable lighting and occlusion, these limitations are even more critical. Thus, while CNNs remain the dominant architecture in plant disease detection, their generalization to fruit diseases in real-world settings remains underexplored.

Eggplant and brinjal disease studies

Most prior work on eggplant has concentrated on leaf diseases. Krishnaswamy and Purushothaman (Krishnaswamy et al. 2020) created a five-class dataset and used VGG16 with multi-class SVM, achieving 99.4% accuracy. Abisha et al. (Abisha et al. 2023) combined Shearlet transforms with CNNs to capture texture and structural cues, reaching 93% mean accuracy. Chelladurai and Sujatha (Chelladurai and Sujatha 2024) proposed DenseNet ensembles on a seven-class dataset of 8,080 leaf images, achieving 94.4%. These studies confirm the promise of CNNs for eggplant leaves, but they rely on controlled leaf images and do not address fruit conditions.

Some studies considered both leaves and fruits. Haque and Sohel (Haque and Sohel 2022) fused CNN-SVM and CNN-Softmax pipelines for nine eggplant diseases, including fruit rot. While accuracy improved over standard models, most of the dataset still comprised leaf images. Venkataramana et al. (Attada et al. 2022) mixed CNNs with SVMs for brinjal disease prediction, reporting 99% accuracy, though validation details were limited. Collectively, these works illustrate the leaf-centric bias in the literature, with fruit disease detection still rare.

Detection and robotics approaches

Object detection methods have been adopted for field use. Nasution and Kartika (Nasution and Kartika 2022) applied YOLOv4 on Raspberry Pi with Telegram notifications, demonstrating feasibility for low-cost, real-time detection. Huang et al. (Huang et al. 2024) introduced YOLOv8-E, enhancing small-object detection and model compactness. Tamilarasi et al. (Tamilarasi et al. 2025) developed YOLOv11s-Brinjal, optimized for harvesting robots, reporting 98% mAP with only 8.2 MB model size. These works highlight YOLO's suitability for real-time detection, yet they target fruit presence or pests, not disease classification on fruits.

Robotics and pest monitoring also appear in parallel. Sutayco et al. (Sutayco et al. 2025) developed a solar-powered rover for insect detection that reached 86% accuracy across classes. While valuable for integrated pest management, these pipelines focus on insect categories rather than disease phenotypes.

Multimodal and advanced sensing

Beyond RGB imaging, multispectral and hyperspectral sensing have been tested. Zhang et al. (Zhang et al. 2024) fused five channels to detect early Verticillium wilt with ~ 87% precision. Wang et al. (Wang et al. 2024) used hyperspectral imaging and autoencoders for leaf nitrogen estimation, achieving $R^2 \approx 0.91$. Huang et al. (Huang et al. 2025) combined VIS-NIRS spectra with CNN+attention+LSTM to predict fruit maturity ($R^2 = 0.876$). These studies prove the potential of spectral signals for plant monitoring but are specialized, expensive, and less accessible for farmer-level solutions.

Trends in architectures

The reviewed studies confirm the dominance of convolutional neural networks and YOLO-based detectors for eggplant disease and pest identification. CNNs, including VGG, ResNet, and DenseNet, have been applied extensively for classification tasks, often achieving high accuracies under controlled conditions (Krishnaswamy et al. 2020; Abisha et al. 2023; Chelladurai and Sujatha 2024). YOLO adaptations have proven effective for real-time detection in field settings, with lightweight variants demonstrating strong potential for mobile and robotic deployment (Nasution and Kartika 2022; Huang et al. 2024; Tamilarasi et al. 2025). Hybrid models that integrate fusion strategies or attention modules have also emerged, but attention has mostly been embedded as a component within CNN frameworks rather than explored through transformer architectures (Haque and Sohel 2022; Wang et al. 2025). Consequently, systematic comparisons between CNNs and vision transformers are still absent in the context of eggplant disease recognition.

Taken together, prior work illustrates notable progress in leaf-based disease detection and object detection pipelines, but fruit disease classification has received little focused attention. While CNNs and YOLO variants dominate the field, issues such as class imbalance, robustness under field variability, efficiency on edge devices, and model interpretability remain insufficiently addressed. These gaps provide the rationale for a comprehensive benchmark of state-of-the-art CNN and transformer architectures for brinjal fruit disease detection under realistic conditions.

Materials and methods

This section describes the methodological framework used in this study which consists of an overview of dataset, preprocessing techniques, experimental setups and the deep learning models applied for brinjal fruit classification. The objective was to enable fair and reproducible comparison between several modern CNN architectures such that the architecture bias can be minimized. To that end, we developed an experimental pipeline which includes baseline benchmarking and hyperparameter optimization as well as a study on advanced training strategies. Flowchart, demonstrating the overall approaches adopted in the present work is shown in Fig. 1 which is organized into three central stages: 1. Data Acquisition & Pre-processing, which includes dataset preparation and balancing; 2. Proposal of an Experimental Framework, showing the multi-phase model assessment (Baseline Benchmarking, Hyperparameter Tuning and Overall Strategy Assessment); and 3. Results & Analysis, where the performance results are discussed and the best model is selected.

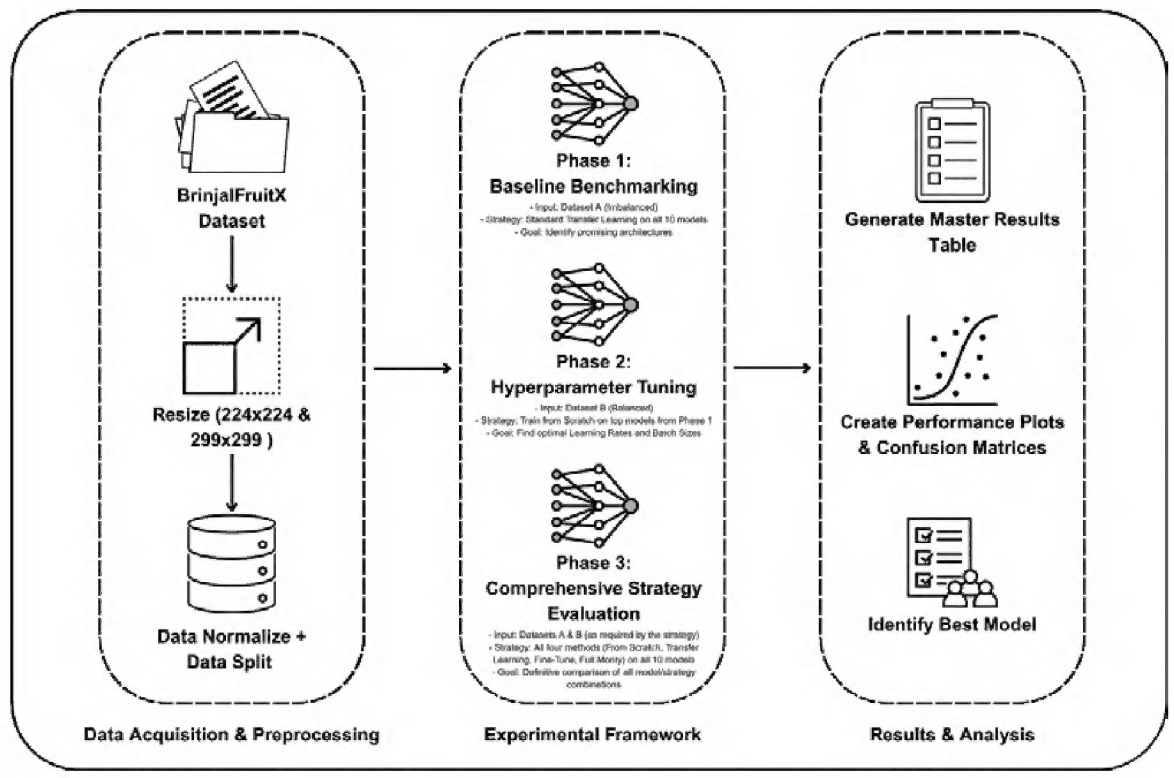


Figure 1. Workflow for brinjal fruit disease classification.

Dataset and preprocessing

In this paper, we employed the BrinjalFruitX (Hasan et al. Bijoy 2025) dataset consisting of brinjal fruit images taken in their original field environments. The data set is representative of real agricultural environment since the images are with different scale, illumination and orientation as well as varying background complexity. This diversity enabled us to build models that are not just accurate but also resilient in the face of real-world challenges. Each photo in the dataset was labelled to various class referring to different state of brinjal fruits. Sample images from all the classes are shown in Fig. 2.

In order to make the dataset ready for training, we introduced a standardized preprocessing pipeline. We began by loading all images with OpenCV image library and then converting the loaded BGR images into an RGB format that are compatible with deep learning tools like TensorFlow and Keras. Each image was then resized to meet the input size of structures as follows: 224224 pixels for MobileNet, VGG, ResNet and EfficientNetB0; 299299 pixels for InceptionV3 and InceptionResNetV2; 600600 pixels (large-scale experiments). This rescaling provided all models with a constant range and allowed the results to be compared. Then we rescaled the pixel values of images from [0, 255] to [0, 1]. This helped to reduce the instability of the gradient updates and increased convergence speed during training.

In this study the dataset was divided into three parts (70% training, 10% validation and 20% test) for experimental consistency. We preserved the class distribution in each split by stratified sampling. We did not use this test set for training or validation at any stage, and used it only for a final unbiased evaluation of model performance. As the dataset showed a modest class imbalance, we took two complementary approaches to deal with it. One way was to keep the original class distribution but to add class weights for loss calculation (Table 2). In the second method, we built a balanced sample by oversampling the minority classes through augmentation such that all categories had around same number of data_points (Table 1). Even though the current study uses publicly available BrinjalFruitX data set where images were already captured in various field conditions. We used data augmentation to

Figure 2. Sample images from the BrinjalFruitX dataset.

Table 1. Class-wise dataset distribution of images before and after oversampling.

Class	Training (Original)	Training (Balanced)	Validation	Testing	Total images (Original)	Total images (New)
Brinjal Fruit Creaking	140	507	20	40	200	567
Healthy Brinjal	359	507	52	103	514	662
Phomopsis Blight	113	507	16	32	161	555
Shoot and Fruit Borer	507	507	73	145	725	725
Wet Rot	141	507	20	41	202	568
Total	**1260**	**2535**	**181**	**361**	**1802**	**3077**

Table 2. Training strategies adopted for model evaluation.

Strategy Name	Initial Weights	Base Model State	Key Techniques Applied
From Scratch	Random (None)	Fully Trainable	End-to-end training from random initialization on the balanced dataset.
Transfer Learning	ImageNet	Fully Frozen	Single-stage training of only the classification head.
Fine-Tune	ImageNet	Stage 1: Frozen Stage 2: Top 50% Trainable	Two-stage training with class weights and a low learning rate during fine-tuning.
Full Monty	ImageNet	Stage 1: Frozen Stage 2: Top 50% Trainable	Similar two-stage training on the balanced dataset, eliminating the need for class weights.

increase real-world generalisation. For the augmentation pipeline we used random horizontal flips, small angle rotations (up to ±20°), zoom in-out variations (±10%) and also brightness-contrast adjustments (±15%), all applied stochastically per mini-batch. Small translation shifts (≤10%) and shear transforms were also applied to mimic random fluctuations in natural fruit orientation and framing across the existing data set. These operations virtually expanded the diversity of training samples and reinforced model's resistance to illumination, orientation and scale changes.

The final class-wise distribution of images across the training, validation, and testing splits is summarized in Table 1. Every row represents the particular brinjal disease class. The columns under Training (Original), Validation, and Testing indicate the dataset distribution before oversampling. The Training (Balanced) column represents the number of samples gained after oversampling the minority class by data augmentation. The Total (Original) column presents the total number of real images per class before oversampling and in Total (New) the total after oversampling. Accordingly, the total number of original and augmented images in the data set was augmented from 1,802 to 3,077.

Training protocols

Stratified sampling was performed while splitting the BrinjalFruitX dataset to obtain 70% training, 10% validation and 20% testing sets for keeping consistency and reproducibility. This helped to maintain the relative distribution of classes and reduced bias toward majority groups. The validation set was used only for hyperparameter tuning/early stopping procedures and the test set remained completely hidden until final evaluation. Pilot 5-fold cross-validation was also performed to check for robustness but results shown in this study are the ones from fixed split, in order to facilitate comparison.

We conduct all experiments on NVIDIA RTX A6000 GPUs. Random seeds were fixed for Python, NumPy and TensorFlow to improve reproducibility. The pre-processing pipeline was common to all models so that the models being compared were trained in identical conditions. Apart from this default protocol, four additional training procedures were implemented to investigate the impact of initialization, pre-training, fine-tuning and dataset balancing. These are presented in Table 2 and an overview of the entire protocol is given in Table 3.

Table 3. Summary of the training protocol used across all experiments.

Parameter	Configuration
Dataset Split	70% training, 10% validation, 20% testing (stratified)
Validation Strategy	Fixed split (pilot 5-fold CV for robustness)
Optimizer	Adam
Learning Rates Tested	1×10^{-3}, 1×10^{-4}, 1×10^{-5}
Batch Sizes Tested	8, 16, 32
Max Epochs	100
Callbacks	EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
Hardware	NVIDIA RTX A6000 GPU
Reproducibility	Fixed random seeds across all libraries

Models selected

In this work, we considered a total of ten convolutional neural network (CNN) architectures from five different families: VGG, ResNet, Inception, EfficientNet and MobileNet. This variety allowed us to perform a thorough comparison of the classical deep networks with recent light-weight architectures focused on efficiency. Representative structures from each of the families are shown in Figs 3–7, and their input patterns are listed in Table 4.

The VGG family (e.g., VGG16 and VGG19) (Simonyan and Zisserman 2015) was one of the earliest successes in

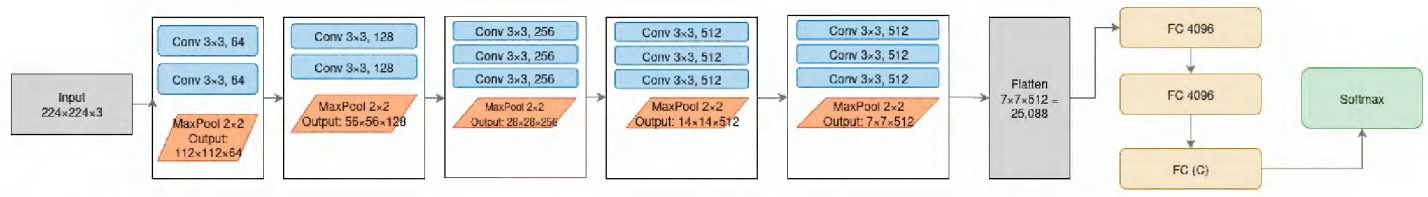


Figure 3. VGG16 architecture with stacked 3 × 3 convolutional layers.

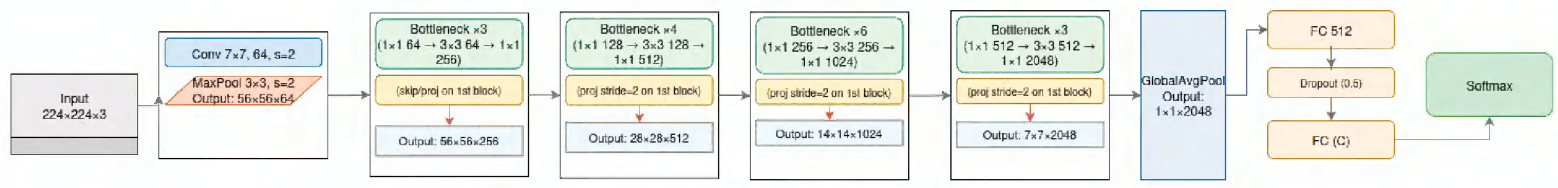


Figure 4. Compact schematic of the ResNet50 architecture.

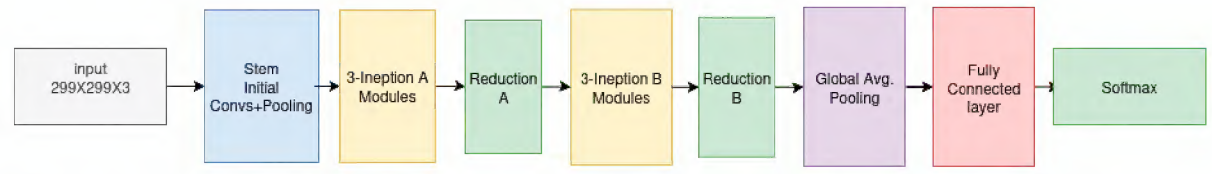


Figure 5. Schematic of the InceptionV3 architecture with inception and reduction modules.

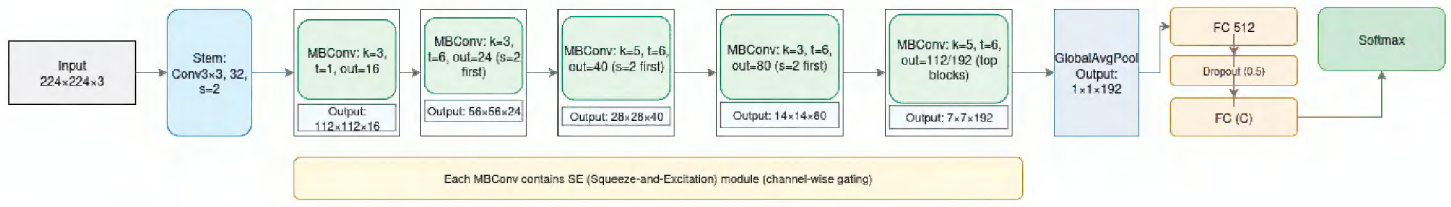


Figure 6. Schematic of the EfficientNetB0 architecture with MBConv blocks.

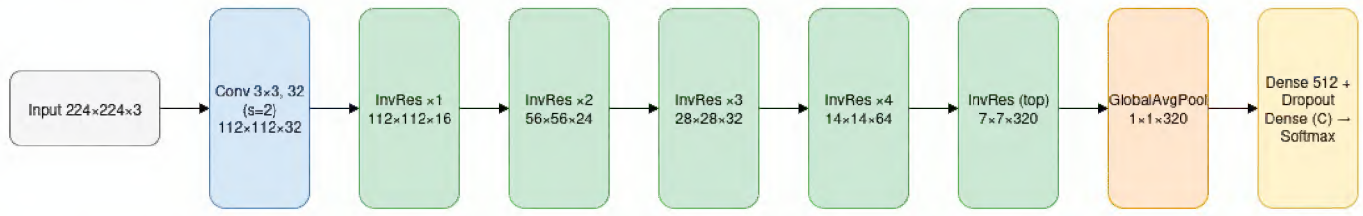


Figure 7. Schematic of the MobileNetV2 architecture with depth wise separable convolutions and inverted residuals.

deep CNN design. These networks consist of sequential stacks of 33 convolutional layers which are followed by ReLU activations and max-pooling layers, and then fully connected for classification. Despite being very computationally intensive in terms of the number of parameters, they are still excellent baselines for extracting features. Fig. 3 is used in the implementation.

The family of ResNets (ResNet50 and ResNet152) (He et al. 2016) presented residual learning by shortcut-connections, which allows very deep models without vanishing gradients. ResNet50 is equipped with 50 layers and relies on bottleneck residual blocks; whereas ResNet152 addresses the representational power issue by scaling the architecture up to 152 layers at the expense of heavier computation. Fig. 4 shows the ResNet50 architecture used in this paper.

The family of inception, such as InceptionV3 and InceptionResNetV2, concentrate on efficient multi-scale feature representation. InceptionV3 (Szegedy et al. 2016) employs factorized convolutions, auxilary classifiers, and reduction modules, whereas InceptionResNetV2 (Szegedy et al. 2017) proposes to stack inception modules with residual connections for further and better model. The trimmed architecture of InceptionV3 in our experiment is shown in Fig. 5.

The EfficientNet (including the smallest model, EfficientNetB0) are a family of models constructed in a similar fashion as follows: balancing network depth, width and resolution (Tan and Le 2019). Leveraging MBConv blocks with squeeze-and-excitation (SE) mechanism, EfficientNetB0 strikes a good balance between accuracy and efficiency. The overview of EfficientNetB0 is presented in Fig. 6.

Last, the MobileNet family (MobileNetV2 and MobileNetV3Large) (Sandler et al. 2018; Howard et al. 2019) is designed to be lightweight. MobileNetV2 utilizes depthwise separable convolutions and inverted residual blocks with linear bottlenecks to reduce computation cost a lot. MobileNetV3Large enhances it with NAS and SE module. The architecture of the MobileNetV2 is represented in Fig. 7.

Table 4 summarizes the models and their standard input sizes used in this study.

Table 4. Summary of CNN models selected for evaluation, showing their optimal input size used in the final experiments.

Model Architecture	Family	Optimal Input Size (pixels)
VGG16	VGG	224 × 224
VGG19	VGG	224 × 224
ResNet50	ResNet	224 × 224
ResNet152	ResNet	224 × 224
InceptionV3	Inception	299 × 299
InceptionResNetV2	Inception	299 × 299
EfficientNetB0	EfficientNet	224 × 224
MobileNetV2	MobileNet	224 × 224
MobileNetV3Large	MobileNet	224 × 224

Experimental environment settings and performance evaluation metrics

All models were implemented in TensorFlow 2.x with the Keras API. We adopted the Adam optimizer, which adaptively tunes learning rates and is widely effective across CNN architectures. Three candidate learning rates (1×10^{-3}, 1×10^{-4}, 1×10^{-5}) were tested; 1×10^{-4} consistently offered the best balance of convergence speed and stability. Batch sizes of 8, 16, and 32 were explored, with smaller values (8 or 16) generally producing better generalization. All models were trained for 100 epochs with common hyper-parameters to allow for direct comparison between architectures. The identical optimization, learning rate schedule, and augmentation pipeline were used consistently across models. We saved model checkpoints after each epoch and then used the best weights according to validation loss for final testing.

The performance of all models was assessed using a comprehensive set of evaluation metrics that provide both overall accuracy and class-level insights. Since the BrinjalFruitX dataset exhibited moderate class imbalance, we emphasized not only accuracy but also precision, recall, and F1-score, along with their macro- and weighted-averaged forms (Sokolova and Lapalme 2009; Powers 2011). In addition, the test loss was reported using sparse categorical cross-entropy (Goodfellow et al. 2016) to capture overall generalization performance.

Classification metrics:

Accuracy was computed as the ratio of correctly classified samples to the total number of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TR, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Precision and recall were calculated for each class as:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

The F1-score, which balances precision and recall, was defined as:

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

To account for multiple classes, we computed two types of F1 aggregation. The macro-averaged F1-score was obtained by averaging the F1-scores across all classes:

$$Macro - Fl = \frac{1}{N} \sum_{i=1}^{N} Fl_i$$

where N is the number of classes. This treats all classes equally, regardless of their frequency. The weighted-averaged $F1\text{-}score$ was also calculated, in which class-specific $F1\text{-}scores$ were weighted by their support (number of true instances per class), thus reflecting the influence of class imbalance.

Loss metric:

The primary loss function was sparse categorical cross-entropy, given by:

$$L_{CE} = -\sum_{i=1}^{N} y_i log\left(\hat{y}_i\right)$$

where y_i is the ground-truth class indicator and $\hat{y}_i$ is the predicted probability for class i (Goodfellow et al. 2016).

Confusion matrix:

In addition to scalar metrics, we also analyzed the confusion matrix (Kohavi and Provost 1998) for each trained model to visualize per-class performance. This provided insights into systematic misclassifications and helped identify visually similar categories where the models struggled.

Efficiency metrics:

To assess the practical viability of each model, we also recorded efficiency-oriented measures including the number of trainable parameters, floating point operations (FLOPs) per forward pass, and average inference time per image (batch size = 1) (Howard et al. 2017; Molchanov et al. 2017). These metrics are crucial for determining the suitability of models for deployment in resource-constrained agricultural settings.

Results and discussion

In this section, we evaluated our approach using the several phase framework described in material and methods section. The main findings are organized in 3 stages of validating results: (1) baseline benchmarking for setting an initial context, (2) hyperparameter tuning for adjusting training dynamics and (3) thorough model/strategy evaluation. In each case, we point out some observations that helped during search and eventually converge to the most effective architecture and training strategy. We have trained all the models with 100 epochs to maintain the balance conditions for all the experiments.

Phase 1- baseline benchmarking

The aim of this phase was to set a baseline of all ten CNN architectures all operating under the same conditions with frozen convolutional weights and a newly trained classification head. This kind of arrangement offered an expedited and consistent place to compare architectural families prior to further optimization. We first conducted a preliminary experiment with all the ten architectures based on simple transfer learning where the convolutional part is frozen and only the classifier is trained. This gave us a quick and stable baseline to compare against. The performance across architectures is summarized in Table 5.

Table 5. Baseline performance of all CNN architectures under transfer learning during Phase 1.

Model	Test Accuracy (%)	Test Loss	Macro-F1	Weighted-F1
MobileNetV2	84.76	0.412	0.846	0.852
InceptionV3	82.45	0.435	0.821	0.829
VGG16	79.32	0.487	0.793	0.801
VGG19	78.94	0.493	0.789	0.798
InceptionResNetV2	77.56	0.521	0.775	0.783
ResNet50	52.08	0.912	0.510	0.523
ResNet152	54.32	0.876	0.529	0.542
EfficientNetB0	40.17	1.134	0.401	0.418
MobileNetV3Large	68.12	0.657	0.673	0.681

Table 6. Effect of batch size and learning rate on selected models during Phase 2.

Model	Batch Size	Acc @ LR = 1×10^{-4} (%)	Acc @ LR = 1×10^{-5} (%)
MobileNetV2	8	91.45	86.32
	16	90.82	85.70
	32	89.54	84.10
InceptionV3	8	88.67	82.93
	16	89.92	84.27
	32	87.54	81.65
VGG16	8	84.23	80.15
	16	85.02	80.74
	32	85.89	81.12

The scores show that smaller, inception-style models (MobileNetV2, InceptionV3) worked better than deeper residual network or EfficientNetB0. Although the performance of ResNet50 and EfficientNetB0 were substandard, MobileNetV2 achieved an accuracy of more than 84% despite having been trained for short duration. These observations led us to choose to concentrate further tuning on the best families. From this simple comparison it can be observed that shallow and inception type models captured discriminative information without being guided by more than just frozen-base transfer learning constraints. This also indicates that MobileNet and Inception like architectures, which feature separate layers taking depthwise separable convolutions or multi-branch receptive fields, are particularly more suitable for the moderate variable nature of BrinjalFruitX. On the other hand, deeper residual/compound scaling models perform worse since they need much larger training sets to converge. The initial results of these bases were used to determine MobileNetV2, InceptionV3, and VGG16 as potentially the most promising architecture to consider tuning later. Based on these preliminary observations, we focused on the MobileNet, Inception and VGG architectures in our experiments.

Phase 2 - hyperparameter tuning

This phase designed to cautiously tweak the training dynamics of our best networks from Phase 1 MobileNetV2, InceptionV3 and VGG16 with a systematic approach that tried learning rates and batch sizes in order to find appropriate configuration that leads into stable convergence/globalization. We conducted a controlled hyperparameter sweep over the best performing models from Phase 1: MobileNetV2, InceptionV3 and VGG16. We also investigated the effects of batch size (8, 16, 32) and learning rate ($1*10^{-4}$ vs. $1*10^{-5}$). Results are shown in Table 6.

Two general findings were robust to all models. First, configurations trained with a learning rate of $1*10^{-4}$ converged faster and at higher peak validation accuracy than those trained with $1*10^{-5}$, which often plateaued at a lower level. Second, smaller batch sizes (8 or 16) led to smoother learning dynamics and better generalization as compared to larger batch sizes (32), which tended to level off at lower accuracy. Critically, the favored setups yielded stable validation plateaus without departing from training curves, reflecting successful generalization and minimal overfitting under our augmentation policy and early stopping regime. In addition to these trends, architectures within each family showed distinct performance behaviours. MobileNetV2 gained a lot from batch size 8 which indicates that diversity in gradients stabilized the depthwise separable layers. InceptionV3 worked best with batch size 16, which is expected given the small overhead it has to compensate for its multi-branch nature. VGG16, on the other hand, was quite insensitive to the batch-size variation but started to show small performance gains as batch-size grew.

These tuning results validated two crucial design decisions going into the next stage: choosing $1*10^{-4}$ as the default learning rate and opting for smaller batch sizes for the MobileNet and Inception families. All the architectures were re-assessed in the four training strategies mentioned above based on these optimized hyperparameters. Using these best settings, we moved to Stage 3 and compared the variants starting from all ten architectures in a carefully organized way for all four training packet strategy.

Phase 3- comprehensive strategy evaluation

This last stage intended to make an "all-ten-architectures-off" benchmarking under the umbrella of four differing training methods (From Scratch, Transfer Learning, Fine-Tune and Full Monty) probing into how the selected learning strategy affects accuracy and generalization ability of models while keeping track on their computational budgets. Following these observations, in the last step we did an extensive benchmarking comparing task performance with respect to how each training strategy affects final accuracy, generalization and computational efficiency among the ten architectures.

There are few observations that can be seen from these findings. First, the strategy selected was found to dramatically impact performance. The Full Monty methodology consistently obtained the best results among strategies for MobileNet, Inception and VGG families of architecture over that achieved by simple transfer learning. Second, performance was widely different across the architectural families: MobileNetV2 (Full Monty)

Table 7. Comprehensive performance evaluation of all models and training strategies.

Model Architecture	Strategy Name	Batch Size	Test Accuracy (%)	Test Loss	Macro Avg F1	Weighted Avg F1	Training Time (HH:MM:SS)
MobileNetV2	**Full Monty**	**8**	**97.98**	**0.0707**	**0.9793**	**0.9799**	**00:18:42**
	Fine-Tune	8	94.35	0.1851	0.9425	0.9438	00:34:12
	Transfer Learning	8	94.76	0.1836	0.9466	0.9478	00:07:17
	From Scratch	8	58.47	0.6789	0.3690	0.4314	00:12:21
InceptionV3	Full Monty	8	**97.58**	0.1027	0.9750	0.9758	00:55:39
	Fine-Tune	8	89.92	0.2717	0.8966	0.8994	00:54:13
	Transfer Learning	8	89.52	0.2724	0.8926	0.8954	00:19:32
	From Scratch	8	93.55	0.2184	0.9328	0.9350	00:44:49
VGG16	Full Monty	16	**97.58**	0.1006	0.9751	0.9758	00:45:12
	Fine-Tune	16	85.89	0.4426	0.8560	0.8594	00:24:44
	Transfer Learning	16	87.10	0.4375	0.8688	0.8717	00:23:49
	From Scratch	16	89.92	0.2788	0.8969	0.8995	01:09:54
InceptionResNetV2	Full Monty	16	96.77	0.0847	0.9668	0.9677	01:55:43
	Fine-Tune	16	83.87	0.3464	0.8383	0.8397	01:40:33
	Transfer Learning	16	84.27	0.3423	0.8420	0.8438	01:05:02
	From Scratch	16	96.77	0.1783	0.9668	0.9677	02:52:20
VGG19	Full Monty	16	95.97	0.1033	0.9582	0.9596	00:55:10
	Fine-Tune	16	85.89	0.4311	0.8516	0.8571	00:30:58
	Transfer Learning	16	87.50	0.4248	0.8680	0.8732	00:30:39
	From Scratch	16	89.52	0.2787	0.8929	0.8955	01:23:02
ResNet50	Full Monty	16	91.13	0.2452	0.9065	0.9101	01:14:26
	Fine-Tune	16	68.15	0.6296	0.6434	0.6631	00:39:31
	Transfer Learning	16	66.53	0.6348	0.6438	0.6586	00:23:34
	From Scratch	16	88.71	0.3176	0.8831	0.8867	00:54:34
EfficientNetB0	From Scratch	16	88.71	0.2548	0.8838	0.8871	00:56:22
	Fine-Tune	16	58.47	0.6829	0.3690	0.4314	00:11:46
	Transfer Learning	16	58.47	0.6851	0.3690	0.4314	00:08:58
	Full Monty	16	64.11	0.6333	0.6300	0.6191	00:16:13
MobileNetV3Large	From Scratch	8	58.47	0.6787	0.3690	0.4314	00:26:39
	Fine-Tune	8	70.97	0.6451	0.6560	0.6790	00:21:25
	Transfer Learning	8	72.98	0.6480	0.6919	0.7102	00:07:39
	Full Monty	8	70.56	0.6484	0.6383	0.6647	00:11:37
ResNet152	From Scratch	16	89.52	0.2791	0.8907	0.8945	02:54:34
	Fine-Tune	16	71.37	0.6304	0.6897	0.7043	01:51:00
	Transfer Learning	16	71.77	0.6271	0.6976	0.7108	01:01:02
	Full Monty	16	87.10	0.2862	0.8655	0.8701	03:45:11

obtained the best trade-off of accuracy and efficiency (97.98% mel-accuracy, macro-F1 = 0.9793) for short training times whereas InceptionV3 and VGG16 also yielded competitive accuracies but relying much higher computations. Third, several families of more complicated models (ResNets, EfficientNetB0) performed worse due to their architectural complexity and also because these models require large datasets to achieve a good score. The overall results are shown in Table 7, indicating test accuracy, test loss, macro and weighted-F1 scores and total training time.

Final efficiency analysis and deployment suitability

This phase measured the computational and deployment efficiency of the best performing architectures with a focus on the trade-off between prediction accuracy, number of parameters and inference latency required for practical applications at field scale. To complement accuracy-based comparisons, we also analyzed model efficiency in terms of parameter count, FLOPs, and inference time (Table 8). Lightweight architectures such as MobileNetV2 (3.4M params, 0.30B FLOPs) provided strong accuracy with a low computational footprint, while VGG variants and InceptionResNetV2 were substantially heavier.

Table 8. Efficiency comparison of the evaluated architectures.

Model Architecture	Parameters (M)	FLOPs (B)	Inference Time (ms)
MobileNetV2	3.4	0.30	3.2
MobileNetV3Large	5.4	0.22	3.5
EfficientNetB0	5.3	0.39	4.1
InceptionV3	23.8	5.7	8.9
InceptionResNetV2	55.9	13.2	15.7
VGG16	138.4	15.3	21.5
VGG19	143.7	19.6	24.8
ResNet50	25.6	4.1	7.6
ResNet152	60.2	11.6	18.3

In order to have a better insight into how training strategy affected learning behaviour, we took an in-depth look at MobileNetV2 across the four training settings.

Convergence dynamics

Transfer Learning Converged quickly initially but it Plateaued quite early because of limited feature adaptation. Training From Scratch was relatively slow and didn't achieve the optimal accuracy because of lack of pretrained knowledge. Fine-Tuning enhanced the generalization by unfreezing the upper layers, enabling deeper refinement of features. Finally, we experimented with the Full Monty strategy, taking its combination of staged unfreezing and balanced data set; it lead the model to have most stable convergence and smooth curve, in which the learned representation was effective, and the optimization procedure was under control.

Per-class behavior

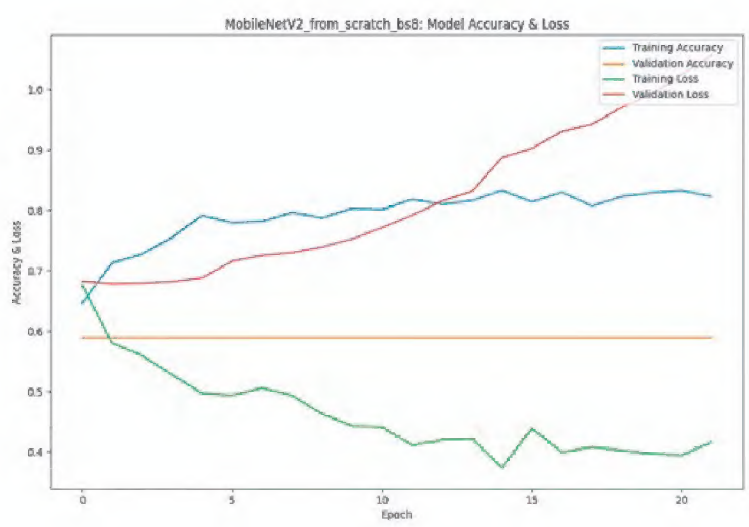
Confusion matrices shown that Full Monty produced the tightest diagonal with minimal off-diagonal errors, whereas Transfer Learning had scattered misclassifications among visually similar classes.

Efficiency vs. performance

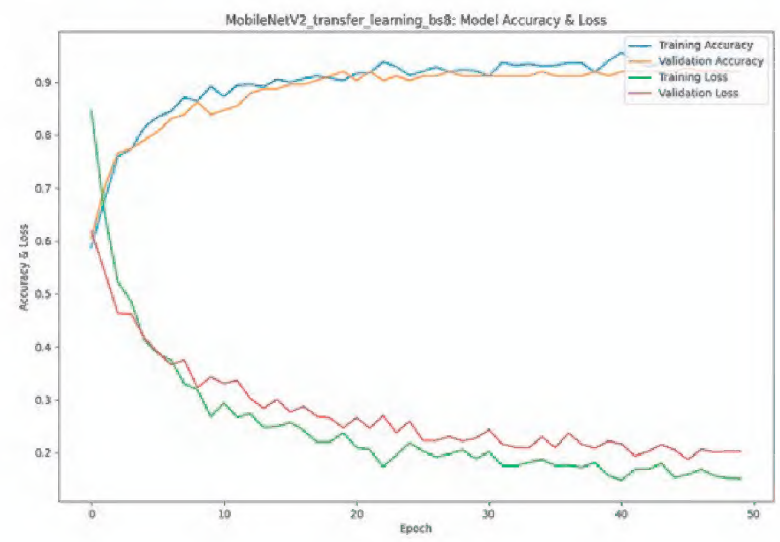
MobileNetV2's near-98% accuracy with a 3.2 ms inference time per image illustrated a favorable accuracy efficiency trade off for field deployment.

These insights are illustrated in Figs 8–11. Fig. 8 compares the training and validation curves of MobileNetV2 under all four strategies. FLOPs are reported for the standard input size of each model (224 × 224 or 299 × 299). Inference times were measured with batch size = 1 on an NVIDIA RTX A6000 GPU. Fig. 9 shows the confusion matrices for MobileNetV2 under Transfer Learning, which shows the class wise prediction behavior. Fig. 10 then compares the confusion matrices of the top-performing models (MobileNetV2, InceptionV3, and VGG16) trained with the Full Monty strategy. MobileNetV2 achieved the cleanest separation, with almost no cross-class errors, while InceptionV3 and VGG16 displayed a few scattered misclassifications.

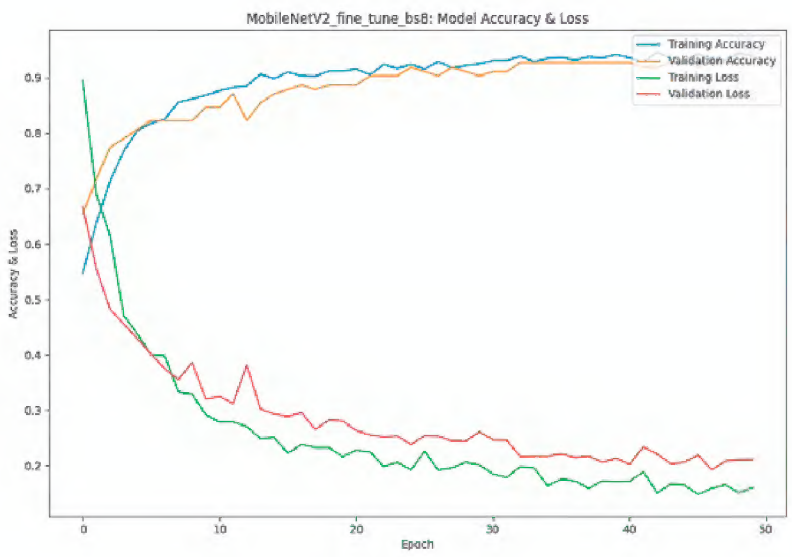
Finally, Fig. 11 presents the ROC curves for MobileNetV2, InceptionV3, VGG16, ResNet50, and EfficientNetB0 under their best-performing training strategies. ROC curves on the test set were calculated, with one-vs-rest strategy. MobileNetV2 achieved the highest AUC, followed closely by InceptionV3 and VGG16, whereas ResNet50 and EfficientNetB0 showed weaker separability. A throughput assessment verified that MobileNetV2 trained using Full Monty offered the best compromise of



(a)From Scratch



(b)Transfer Learning



(c)Fine Tune

(d) Full Monty

Figure 8. Training and validation curves for MobileNetV2 under four strategies on test set.

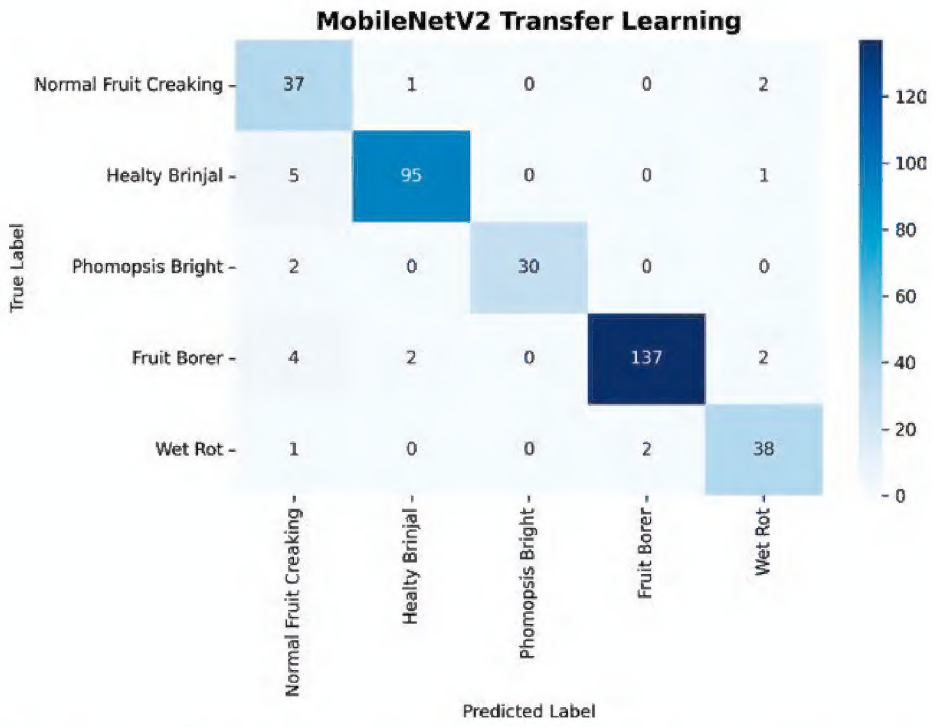


Figure 9. Confusion matrices for MobileNetV2 under transfer learning on test set.

This slower unfreezing strategy helped the models to adapt better to BrinjalFruitX dataset's features. However, poor performance was observed when training completely From Scratch (for all architectures except for InceptionV3, it showed some ability to learn without pre-trained weights). The experiments further showed significant disparities between the model families. MobileNetV2 was the architecture that emerged as the most promising one due to a trade-off between high accuracy and speed training along with well balanced per-class performance. InceptionV3 and VGG16 were also high accuracy, but took 2 3 times longer to train. VGG19, having more parameters, had the same training time. In contrast, ResNets of greater depths and EfficientNetB0 did not generalize well. So even having been trained with m s t best strategy these models plateau at < 92% accuracy that converges much slower.

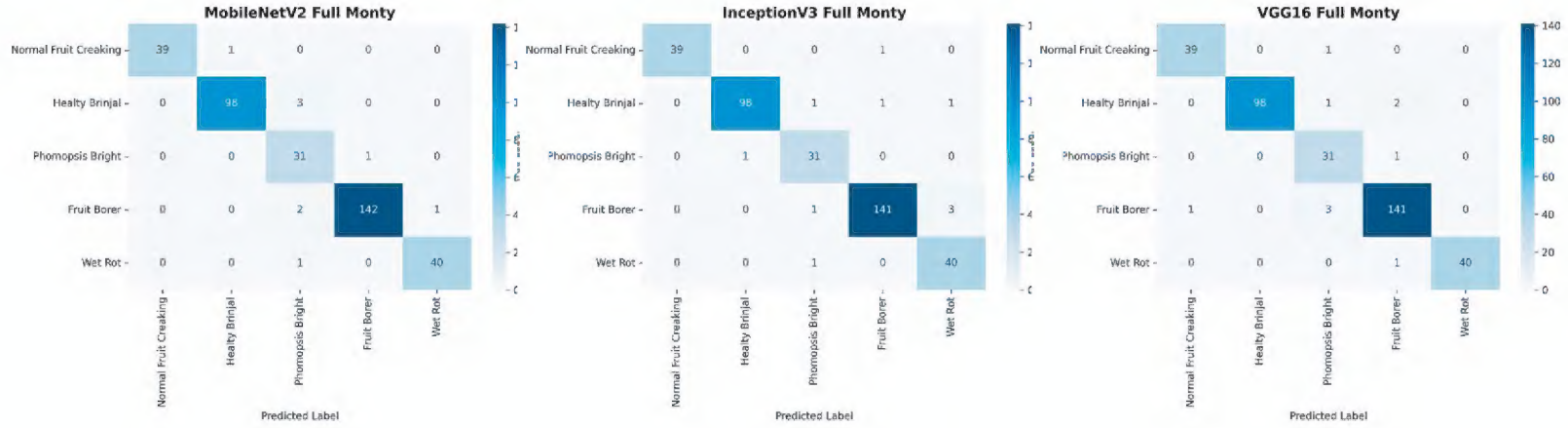


Figure 10. Confusion matrices for top-performing models under the Full Monty strategy on the test set.

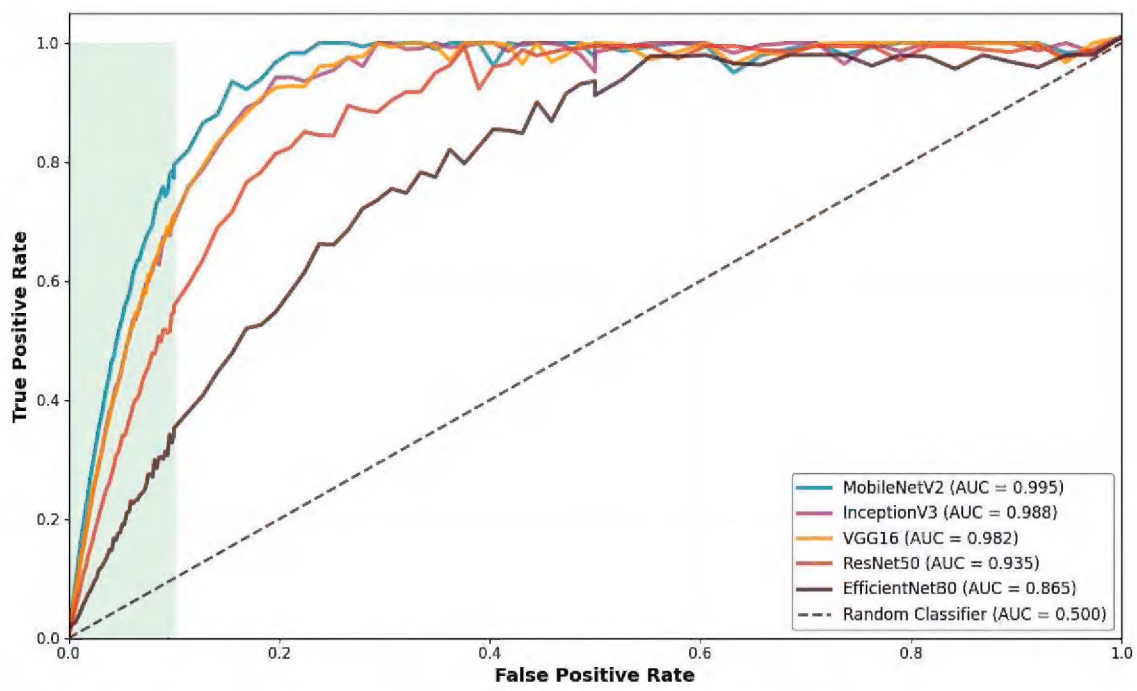


Figure 11. ROC curves for representative models under their best strategies.

accuracy, processing and inference latency for real-time on-device adoption/suitability in agricultural areas.

Discussion

Results over the three experimental phases make it clear that training policy had an important effect on performance. For all four strategies tested, Full Monty consistently resulted in the most superior outcomes. This indicates that deeper or more complex models do not necessarily mean better performance, especially in the case of moderate-size and class-diverse datasets.

An efficiency analysis will give a different view. MobileNetV2 performed close to a 98% accuracy with only 3.4 million parameters and for 0.30 billion FLOPs, took an image of 3.2 ms whereas VGG16 required more than 138 million parameters for over 15 billion FLOPs took an average time of 21.5 ms per image but did not provide any real benefit in terms of accuracy. InceptionV3 also reached a high accuracy

and but required almost one order of magnitude FLOPs compared to MobileNetV2. ResNet152 and InceptionResNetV2 were the heaviest ones, that trained for many hours and needed to wait long time for inference while not doing much better than lighter models. These results underscore that the tradeoff between accuracy and efficiency cannot be generally treated separately—especially for practical agricultural applications, where speed of solutions and resource limitations are as important as incremental performance.

Taken together, the study demonstrates that MobileNetV2 with the Full Monty strategy is the most effective choice for the classification of brinjal fruits. fruits. It offers a strong balance of accuracy, efficiency, and robustness, making it well suited for real-world use. Looking ahead, future work could explore transformer-based models, ensemble methods, or larger field-collected datasets to further improve generalization while maintaining the balance between performance and efficiency.

Conclusion

This work introduces a benchmark of state-of-the-art CNN and transformer-based architectures for the classification of brinjal disease on fruits using the BrinjalFruitX dataset under realistic field conditions. This study evaluated the different transfer learning strategies, batch size variations and training dynamics across ten CNN architectures using both original and augmented dataset. Standardizing the prototyping pipeline and evaluating four training schemes, we achieved good result on predictive performance and computational efficiency. Performance was highly dependent on the training strategy, as indicated by the outcomes. It is also observed that the full Monty approach which was trained on a balanced test set performed best. We found an excellent performance of MobileNetV2 model with test accuracy higher than 98%, and its macro- and weighted-F1 scores were obtained at 0.98, indicating that only 3.4M parameters, 0.30B FLOPs and time in the range of 3.2 ms per image are required. This configuration achieves correctness and efficiency, which can be deployed in agriculture-limited resources. InceptionV3 and VGG16 were also effective but at a high computational cost, whereas deeper residual- and compound-scaling-based models including ResNet and EffcientNetB0 under-performed. Taken together, lightweight and well-tuned CNNs still represent the most feasible choice for fruit disease classification. While the MobileNet V2 model achieved an overall good performance, the experiment only considered one region-of-interest (ROI)-specific dataset and CNN-based networks. It also only concentrated on static RGB imagery, and did not consider multimodal or temporal information. Future work should explore transformer-based models, multimodal sensing (e.g., RGB + spectral imaging), and larger cross-regional datasets to strengthen generalization. Field trials and deployment on mobile or robotic platforms will also be vital to translate these advances into farmer-ready tools for sustainable plant health monitoring.

Authors contribution

BN and YG conceptualized the study and were the supervision of the research as a whole. ST performed the preparation of data and preprocessing as well as implemented the experiment. Analysis and drafting of the manuscript were done by BN and YG. YG and MA revised the results, edited and validated them. The final manuscript was read and approved by all the authors.

Funding

This work was supported by the Deanship of Scientific Research, the Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia, under Grant No. KFU253803

Data availability

The BrinjalFruitX dataset used in this study is publicly available on Mendeley Data at https://doi.org/10.17632/ngc58fsxgd.1. Code supporting the experiments can be found at https://github.com/SAGAR-TAMANG/ADLMODCT.

Declaration

The authors declare that they have no competing interests.

References

Abisha S, Mutawa AM, Murugappan M, Saravanan KS (2023) Brinjal leaf diseases detection based on discrete shearlet transform and deep convolutional neural network. PLoS ONE 18(4): e0284021. https://doi.org/10.1371/journal.pone.0284021

Afshin A, Sur PJ, Fay KA, Cornaby L, Ferrara G, Salama JS, Mullany EC, et al. (2019) Health effects of dietary risks in 195 countries, 1990–2017: A systematic analysis for the global burden of disease study 2017. The Lancet 393: 1958–1972. https://doi.org/10.1016/S0140-6736(19)30041-8

Alkanan M, Gulzar Y (2024) Enhanced corn seed disease classification: leveraging MobileNetV2 with feature augmentation and transfer learning. Frontiers in Applied Mathematics and Statistics 9: 1320177. https://doi.org/10.3389/fams.2023.1320177

Attada VR, Suresh K Kumar, Suganthi N, Rajeswari R (2022) Prediction of brinjal plant disease using support vector machine and convolutional neural network algorithm based on deep learning. Journal of Mobile Multimedia 18: 771–788. https://doi.org/10.13052/jmm1550-4646.18315

Chelladurai K, Sujatha N (2024) Ensemble of densely connected convolutional networks for brinjal leaf disease detection. International Journal of Intelligent Systems and Applications in Engineering 12: 676–683.

Food and Agriculture Organization of the United Nations (2022) World Food and Agriculture – Statistical Yearbook 2022. FAO. https://doi.org/10.4060/cc2211en

Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press. http://www.deeplearningbook.org

Gulzar, Y (2024) Enhancing soybean classification with modified inception model: a transfer learning approach. Emirates Journal of Food and Agriculture 36: 1–9. https://doi.org/10.3897/ejfa.2024.122928

Gulzar, Y(2025a) Papaya leaf disease classification using pre-trained deep learning models: A comparative study. Applied Fruit Science 67: 1–10. https://doi.org/10.1007/s10341-025-01533-1

Gulzar, Y (2025b) PapNet: An ai-driven approach for early detection and classification of papaya leaf diseases. Applied Fruit Science 67(4): 1–11. https://doi.org/10.1007/s10341-025-01466-9

Gulzar Y, Ünal Z, Aktaş H, Mir MS (2023) Harnessing the power of transfer learning in sunflower disease detection: A comparative study. Agriculture (Switzerland) 13(8): 1479. https://doi.org/10.3390/agriculture13081479

Haque MR, Sohel FA (2022) Deep network with score level fusion and inference-based transfer learning to recognize leaf blight and fruit rot diseases of eggplant. Agriculture (Switzerland) 12(8): 1160. https://doi.org/10.3390/agriculture12081160

Hasan, MdZ, Bitto AK, Bijoy MdHI (2025) BrinjalFruitX: A field-collected image dataset for machine learning and deep learning-based disease identification in brinjal fruits. Mendeley Data, V1. https://doi.org/10.17632/ngc58fsxgd.1

He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90

Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, Wang W, Wang W, Chen L-C, Tan M, Chu G, Vasudevan V, Zhu Y, Pang R, Adam H, Le Q (2019) Searching for MobileNetV3. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

Howard A, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In arXiv Preprint. https://arxiv.org/abs/1704.04861

Huang B, Xue L, Yang J, Yin C, Liao K, Liu M, Li J, Shang M (2025) Regression study on fruit-setting days of purple eggplant based on in situ VIS-NIRS and attention cycle neural network. Journal of Food Science 90. https://doi.org/10.1111/1750-3841.17593

Huang Y, Zhao H, Wang J (2024) YOLOv8-e: An improved YOLOv8 algorithm for eggplant disease detection. Applied Sciences (Switzerland) 14(18): 8403. https://doi.org/10.3390/app14188403

Kohavi R, Provost F (1998) Glossary of terms. Machine Learning 30: 271–74 [2-3]. https://doi.org/10.1023/A:1017181826899

Krishnaswamy Rangarajan A, Raja P (2020) Disease Classification in Eggplant Using Pre-Trained VGG16 and MSVM. Scientific Reports 10. https://doi.org/10.1038/s41598-020-59108-x

Molchanov P, Tyree S, Karras T, Aila T, Kautz J (2017) Pruning Convolutional Neural Networks for Resource Efficient Inference. In International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1611.06440

Mvungi HHA, Wanyama R, Dinssa FF, Ngoteya G, Sigalla J, Minja R, Bishop G, Schreinemachers P (2025) Production of african eggplant in tanzania: varieties, practices, profitability, and constraints. Frontiers in Sustainable Food Systems 9: 1461851. https://doi.org/10.3389/fsufs.2025.1461851

Nasution SW, Kartika A (2022) Eggplant disease detection using yolo algorithm with telegram notification. International Journal of Engineering, Science and Information Technology 2: 127–132. https://doi.org/10.52088/ijesty.v1i4.383

Powers DMW (2011) Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. Journal of Machine Learning Technologies 2(1): 37–63. https://arxiv.org/abs/2010.16061

Professor Jayashankar Telangana State Agricultural University (2025) Summer 2024–25 Pre-Harvest Price Forecast of Brinjal. https://www.pjtau.edu.in/files/AgriMkt/2025/February/Brinjal-Yasangi-Forecast-February-2025.pdf

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Seelwal P, Dhiman P, Gulzar Y, Kaur A, Wadhwa S, Onn CW, Goyal N, Shanmugasundaram H (2024) A systematic review of deep learning applications for rice disease diagnosis: Current trends and future directions. Frontiers in Computer Science 6: 1452961. https://doi.org/10.3389/fcomp.2024.1452961

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1409.1556

Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Information Processing & Management 45(4): 427–37. https://doi.org/10.1016/j.ipm.2009.03.002

Sutayco MJY, Dela ES Cruz, Aranza H, Collado GF, Lui KB (2025) Development of a non-destructive robot for eggplants (solanum melongena) pest detection and classification. Proceedings on Engineering Sciences 7: 1325–34. https://doi.org/10.24874/PES07.02C.013

Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-V4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI Conference on Artificial Intelligence 31(1): 4278–4284. https://doi.org/10.1609/aaai.v31i1.11231

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Tamilarasi T, Muthulakshmi P, Miraei Ashtiani SH (2025) Improved YOLO-based real-time brinjal detection algorithm for vision modules in harvesting robots. Engineering Research Express 7: 035234. https://doi.org/10.1088/2631-8695/adef00

Tan M, Le Q (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), 6105–6114. https://arxiv.org/abs/1905.11946

Wang H, Wei Z, Yang Y, Hou J, Sun ZL, Hu J (2024) Estimation of eggplant leaf nitrogen content based on hyperspectral imaging and

convolutional auto-encoder networks. Spectroscopy and Spectral Analysis 44: 2208–2215. https://doi.org/10.3964/j.issn.1000-0593(2024)08-2208-08

Wang X, Yan F, Li B, Yu B, Zhou X, Tang X, Jia T, Lv C (2025) A multimodal data fusion and embedding attention mechanism-based method for eggplant disease detection. Plants 14(5): 785. https://doi.org/10.3390/plants14050785

World Health Organization (2025) Anaemia in Women and Children. https://www.who.int/data/gho/data/themes/topics/anaemia_in_women_and_children

Zhang Y, Zhang D, Zhang Y, Cheng F, Zhao X, Wang M, Fan X (2024) Early detection of verticillium wilt in eggplant leaves by fusing five image channels: A deep learning approach. Plant Methods 20: 173. https://doi.org/10.1186/s13007-024-01291-3